DEVELOPMENT OF SOFTWARE FOR DIFFERENT TEXTURED PROFILE THRUST BEARING

A Project Report Submitted In Fulfillment Of B. Tech.
In
Mechanical Engineering

By

Name: **Bijay Kumar Kawar**
Roll-111ME0301

Under The Guidance of
Prof. Suraj Kumar Behera

Department of Mechanical Engineering, National Institute of Technology Rourkela

NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA

CERTIFICATE

This is to certify that the project entitled "**DEVELOPMENT OF SOFTWARE FOR DIFFERENT TEXTURED PROFILE THRUST BEARING**", submitted by Mr Bijay Kumar Kawar (Roll No.- 111ME0301), is a record of bona fide research carried out by him at our institute, National Institute of Technology, Rourkela, under my guidance and supervision. The work incorporated in this project has not been, to the best of my knowledge, submitted to any other University or Institute for the award of any degree or diploma.

Date:
Place:

Prof. Suraj Kumar Behera
Asst. Professor
Department of Mechanical Engineering
NIT ROURKELA
Odisha-769008

ACKNOWLEDGEMENT

It provides me immense joy to express my earnest feeling of appreciation to my supervisor **Prof. S.K. Behera**, for his significant inspiration, consistent spark, direction more just for his ever co-working demeanor that encourage me to progress this project in the present form. It was the shot of a life time to interact with an identity that is a perfect mix of head and heart, dynamism and delicacy, wit and gravity with judiciousness and curiosity. I record my unspeakable appreciation for him.

I am also grateful to **Prof. S.S. Mahapatra**, Head of the Department, Mechanical Engineering and **Prof. S. K. Sahoo and Prof. S. C. Mohanty,** Course Coordinator for their help and advice during the course of this work.

I am greatly thankful to my seniors, colleagues, friends for their support and help and finally to my parents for their constant blessing and motivation without which nothing would have been possible.

Date:
Place:

Bijay Kumar Kawar
Roll-111ME0301
Bachelor of Technology
Dept. of Mechanical Engineering
NIT Rourkela
Odisha -769008

ABSTRACT

We know friction is a necessary evil. Friction causes wear and tear. In the field of thrust bearing researchers are finding new ways to reduce friction and hence wear and tear. One of the methods is to use texture over the thrust bearing which will reduce the area of contact between two surfaces and hence it will reduce the friction coefficient and wear and tear of the acting surface.

In forming the texture, regular micro structure were formed on the surface in the form of dimples or bumps. Numerical analysis was carried out before by the researchers to know the properties of spherical, cylindrical, elliptical, rectangular and square texture on the thrust bearing. In the present work, software of the interface was developed to find out the different behavior shown by spherical, cylindrical, elliptical, rectangular and square texture on the thrust bearing. With the help of etching process and laser technology it is very easy to fabricate these kind of five texture economically, so these kind of texture were chosen for thrust bearing. Previously researchers have done the numerical analysis to know the properties of thrust bearing by varying the parameters like aspect ratio, number of rows and columns, number of texture on both circumferential and radial directions on thrust bearing. By using this software researchers from around the world can study the various effects of texture profile without doing any rigorous analysis and it will save a lot of time for them. The software will be capable of calculating pressure profile and load carrying capacity by providing the input parameter of the texture height. An effort is made to develop interactive and user-friendly software capable of evaluating and generating different graphs to further the knowledge on different textured thrust bearing.

CONTENTS

LIST OF FIGURES

CHAPTER 1

1.1 INTRODUCTION

Tribology is a new and exciting field of mechanical engineering. Basically studies on lubrication, friction, wear and tear and contact mechanics comes under its purview. Study of tribology helps us in reducing the maintenance costs of material, increasing the life span and reliability and wear and tear between the equipment like daily use cheap equipment to more complexes and specialized equipment for specific task. It can be further realized that to do a successful design, knowledge of tribology is very much necessary in getting the whole picture about the relative motion between the mating surfaces and design bottleneck.

Due to ever changing design parameters of rotors, different types of bearings have to be designed keeping those rotors in mind. Design limitations like load carrying capacity of journal and thrust bearings create a lot of problem if not tackled very carefully. Wedging action create pressure on the bearings. An elegant solution to this would be to modify the geometry of the bearing surface. In tribology, the very concern is the reduction of friction at any given situation using different technologies right from design to fabrication/manufacturing. At design level different kind of geometry are formed to reduce the contact and doing away the application of lubricants. In manufacturing, micro fabrication techniques are implemented which can form dimple or bump at a very minute scale on the surface of the bearing. In application like mating of surfaces these steps overall lead to drastic change in wear and tear and hence increasing in lifespan of precious equipment. In current scope of the project, software of the interface for different textured thrust bearing is developed with the knowledge from the previous researchers about the different textured thrust bearings.

It can be validated that pressure profile, load carrying capacity are very important design parameters. So, interactive and user-friendly software capable of evaluating and generating different graphs to further the knowledge on different textured thrust bearing is developed with the help of MATLAB software environment in general and GUIDE module in particular.

1.2 AIM OF THE WORK

Development of user friendly, intuitive software to find pressure profile, load carrying capacity as well as different plots between them and the 3-dimensional view of the different profile textured thrust bearing.

CHAPTER 2
LITERATURE SURVEY

Recent and past studies have study the effects of these different methods and of various micro asperity parameters on Coefficient of friction, wear, and load carrying capacity.

Durjyodhan Sethi et al [1] have done a very tremendous work in automating the analysis of the different textured thrust bearing with the help of MATLAB. He studied the basic governing principles of textures on the thrust bearing surface and applies the knowledge to program a code to evaluate different textures. Basically the project has done the all ground works and complete all the modeling through code which helps any researcher to not go through the basic principles of the textured thrust bearing to waste time rather focus on designing required thrust bearing with right texture. All the textures examined by him have various properties which can be assembled together to have them in a single user interface for faster decision making.

Siddhartha Meher et al [2] have done the rotor dynamic analysis of flexible rotor. He examines the different software development model and finds the pros and cons of every model. He has also done the numerical analysis and through GUIDE iterated the design through various stages. This knowledge helps us keep track of how to develop a front end with various parameters to change for easy simulation and decision making on a rotor for specific application. He has done some software testing through different approaches. These testing have been described very nicely with examples which show us how to debug a code without any error.

Edelhofer et al [3] has done research on deploying and sharing of application in the platform of MATLAB. He commented on different strategy for deployment of codes by breaking it further into subprojects and merges them through a different program/ user. This paper guides us through the procedure of choosing the best deployment approach for MATLAB application. He stresses on the fact that we have to understand the application structure before deploying any strategy. Front end and back end are the two parts of a MATLAB programming language. Front end is in direct contact with the user and it determines the user experience through the control of different control buttons or boxes. Back end do all the calculations and computing things and shows the results as graphics on the screen. Front end part needs no special attention whereas

back end determines whether the software will work correctly or not as it has to interact very seamlessly with the front end to give a very good user experience.

Sarah Wait Zaranek, Bill Chou, Gaurav Sharma, and Houman Zarrinkoub, et al [4] wrote an article on how to decrease the time to process the applications and algorithms through different tricks and techniques. Code analyzer and MATLAB profiler, these two tools helps in faster writing of codes without any error. Code analyzer tracks the error in runtime and helps us identify any potential problem every other time. It can work on whole document or a part of it as desired. Code analyzer helps us in maintaining and improving the performance of the code written. The MATLAB profiler helps us in tracking where the code is getting slow and where it working absolutely fine. It gives us in a summarize manner so as to assess the code with great ease. With the help of the report we can find the bottle neck in the code and change it to efficiently run the code. They also advise us on how to use less memory and computing power to run a code using different techniques.

Rob Purser et al [5] have done a project on how to do measurement using MATLAB program. His project gives us an idea as how to develop software from idea to implementation. His way of doing the project has given us the knowledge to work on the project with confidence and a clear road to approach the project. The problem of manual testing is overcome by the use of GUI in MATLAB. In our project this knowledge will be helpful in creating GUI to calculate different properties of textured thrust bearing without any manual or tedious work of numerical analysis of ever changing parameters.

Cleve Moler et al [6] have tried to simulate the famous game of black jack using the MATLAB. Basic strategy of the game was developed way before and with this knowledge he tries to simulate all the possible situations very efficiently in MATLAB. It is a project where idea was there before and with the help of simulation the idea was computed in the MATLAB and this helps in gaining advantage while playing a game. The project overall takes us through the basic strategy to the fine details of the game towards the implementing through code.

CHAPTER 3

GUI in MATLAB

3.1 Introduction

MATLAB (matrix laboratory) is a fourth-generation programming language. It is developed by MathWorks. We can do these following things in MATLAB.

- Matrix initialization and manipulations
- 2-dimentional and 3-dimensional graph plotting
- Creating user interface through GUIDE module
- Doing simulations through SIMULINK module
- Code compatibility with C, and C++

MATLAB is complete software for writing codes, editing codes, debugging of codes without going anywhere else. It also supports object-oriented programming which is also a feature of C++ programming language.

MATLAB is widely used in engineering society as well as in the researchers' community to develop model or creating user interfaces for its data structure capability and editing functions.

3.2 Steps to open GUI in MATLAB

- Open MATLAB by double left click on the MATLAB icon
- Left click on New in the "Menu Bar"
- Left click on "Graphical User Interface"
- Left click on "Create New GUI"
- Left click on "Blank GUI(Default)"
- Left click on ok

3.3 GUIDE – GUI Development Environment

GUIDE is the development environment module for Graphical User Interface in the MATLAB programming language. GUIDE is divided into different categories to develop the user interface.

Components of this environment are

- Menu Bar
- Tool Bar
- Function/Control Bar
- Work Area

3.4 User Environment

There is all total 14 feature function to use from the function bar area. They are

- Select
- Push Button
- Slider
- Radio Button
- Check Box
- Edit Text
- Static Text
- Pop-up Menu
- List Box
- Toggle Button
- Table
- Axes
- Panel
- Button Group

Tool bar contains features like

- Figure activator
- Object browser
- Property inspector
- Menu editor

- Alignment tool
- Undo
- Redo
- Cut
- Save

Menu bar contains features like

- File
- Edit
- View
- Layout
- Tools
- Help

Work area or layout area is given to do the required interface preparation to do a certain task. Features from tool bar and function bar helps to create a UI from inception or idea stage to the design stage. It is very user friendly and has features which are easy to use and manipulate at any instant. It has in-built debugging tool which takes care of the error and notifies when there is one.

3.5 Steps followed to create a GUI:

For creating a complex and complete GUI, we can divide the whole work into three stages. They are

- Designing the GUI
- Laying out in the GUI environment
- Programming the callbacks of the GUI

It is often advised to design a GUI on a sheet before moving out straight to the implementation stage. It helps in realizing the fault in the design to rectify before doing any major work. It also helps unleash the creativity in the design process which is often missing in the implementation phase. The implementation phase only helps in bringing incremental change rather than any fundamental change to the overall design.

After validating the design on paper through brainstorming and iterations it is the stage of laying out the GUI in the development environment. We design the GUI in the work area with the help of feature in tool

bar and function bar. By using the push button and radio button we can have a selection. Through edit text we can put value in the run time as an end user. Static text has the feature of showing message to guide us through the UI. Axes show us the graphs and plots as output in the area provided. After fully laying out the GUI it is the turn to change the string and most importantly the tag of the function. It can be done by double clicking on the function and changing it when required. This tag property is very much essential to code the GUI program as it helps to access the feature through different sub-functions in the code.

After laying out the GUI it is the turn to save the layout which creates two kinds of file with different extensions like .fig file and .m file. .fig file contains the layout of the GUI and .m contains the all the empty callbacks and we need the respective codes to run the GUI. It is very much necessary to code these callbacks because only with the layout it is impossible to run it in a desired manner. Callbacks can use the tag property of the functions to create a meaningful way of combining the functions to work in unison to perform the specific task.

CHAPTER 4

GUI FOR TEXTURED THRUST BEARING

4.1 Introduction

Our aim is to create a GUI which will take the following inputs.

1. Hg : Texture Height
2. AR: Aspect Ratio
3. nx: No of meshes in X direction
4. nr: No of meshes in R direction
5. NC: No of columns of texture
6. NR: No of rows of texture

Our output will show these following things.

1. Pressure Profile
2. Load Carrying Capacity

In addition to this we also have to plot some graphs between different parameters like load vs aspect ratio, friction vs aspect ratio, load vs texture height, friction vs texture height, pressure profile.

We would build the GUI in several steps from a simple GUI with less no of choices to a full scale GUI for entering different parameters. In the first stage we would concentrate on only the spherical textured thrust bearing having positive asperity. We would produce the plot of only the pressure profile. After having some expertise in this environment then we would proceed step by step to complete the GUI.

4.2 Software for Different Textured Profile Thrust Bearing

For textured thrust bearing UI we have different components at initial stage. They are

- Type of Texture

- Asperity
- Texture Dimension
- Grid Dimension
- Parameters
- Output Style
- Instruction Box
- Output Area
- Plot Controls

Under Type of Texture, we have

- Spherical
- Cylindrical
- Elliptical
- Rectangular
- Square

We can select anyone from this group but not more than one.

Under Asperity, we have

- Positive
- Negative

Here also we can select any option but not more than one.

Under Texture Dimension, we have

- No of Rows
- No of Columns

We have to give input both to no of rows and no of columns and not be left blanked to produce an output.

Under Grid Dimension, we have

- Nx
- Nr

Here we have to give input to both radial and axial dimension for output to be shown.

In Parameters, we have pressure profile and load carrying capacity. In both options we have to give the input value.

After giving all these input we then have the options to click like calculate and plot. We can click according to our requirement.

In output style components, we have to choose between model and graphs according to the requirement.

In the Instruction Box, it will be shown whether the inputs are valid or not and other essential information for the comfort of the user.

In the Output Area, we have areas dedicated to graph plot and plot controls.

In Graph Plot Area, the output graph will be shown. In the plot controls, we have the option to input the title of the graph, X-axis, Y-axis control functions. Then we have the comfort to update, save plot and print options.

At last we have two options.

- New
- Close

By clicking New, we can again input another set of data and close button will close the UI environment.

4.3 Iterations

There are up to 4 iterations for the front ends.

- From 1st to 2nd iteration, we have representation of grid dimension that have Nx and Nr for the input, comes in the control flow after the texture dimension. Also the output style goes just before instruction box. We have also slight repositioning of the axes in the output area.
- In 2nd to 3rd iteration, we have eliminated the calculate push button and repositioned the plot after the output style.
- In 3rd to 4th iteration, we have repositioned the parameters viz pressure profile and load carrying capacity to the output style and introduced the input parameter under which we have texture height. In this last iteration, we have also test cased the axes with another simple front end which works fine.

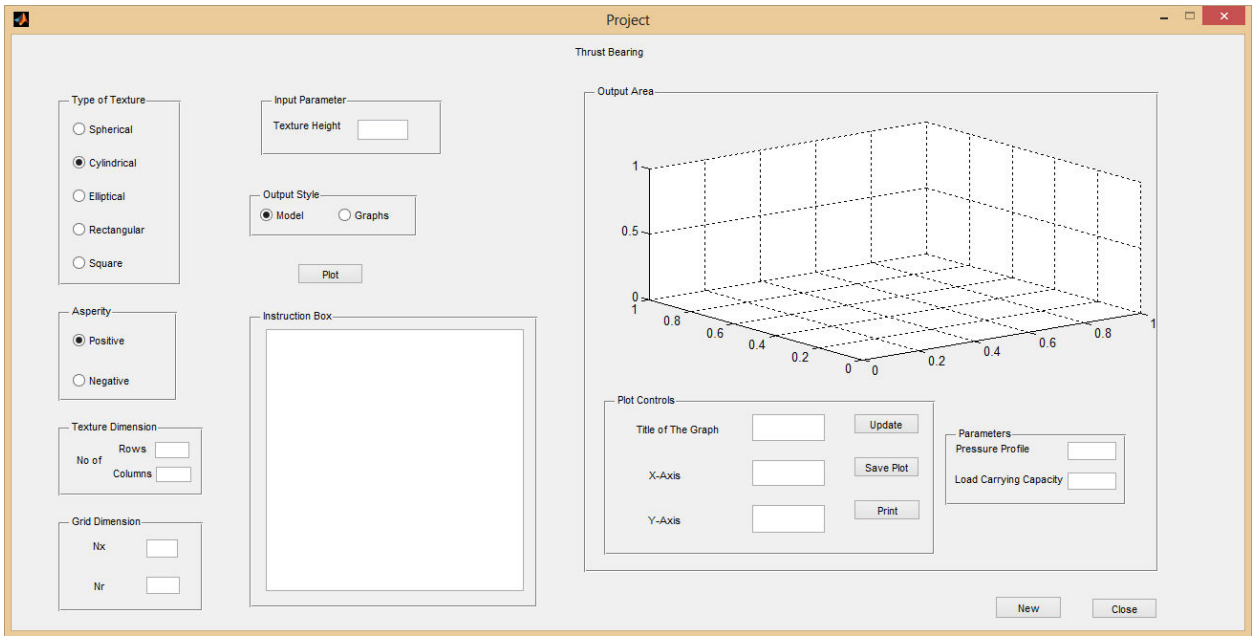


Figure-1: Iteration 1

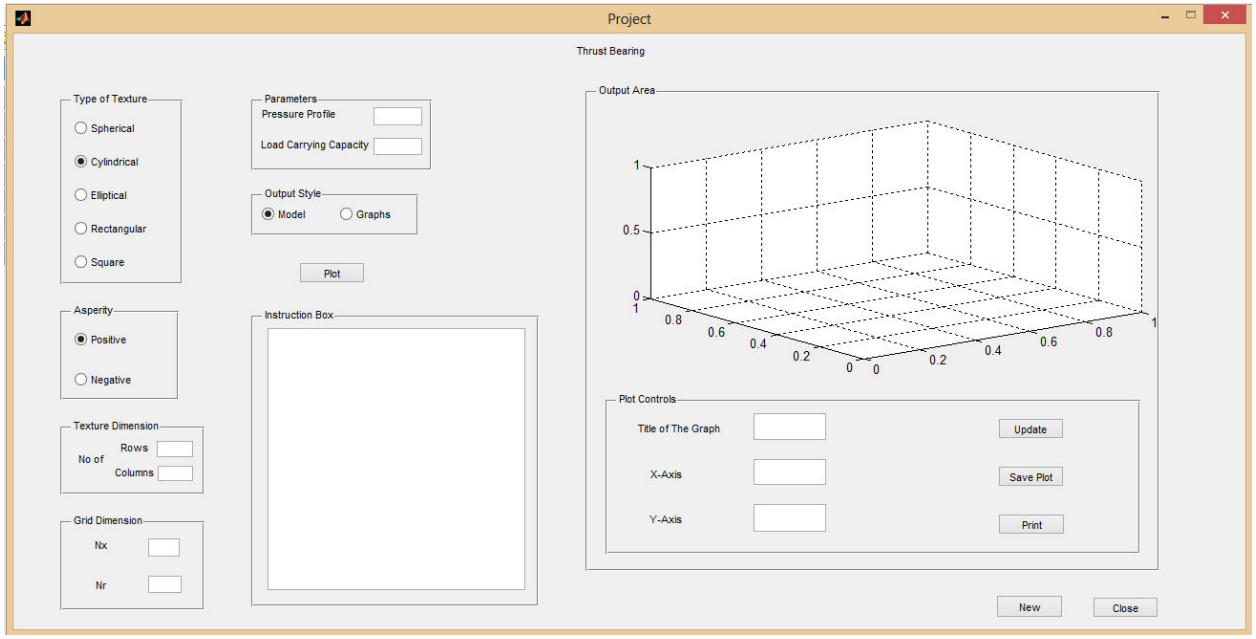


Figure-2: Iteration 2

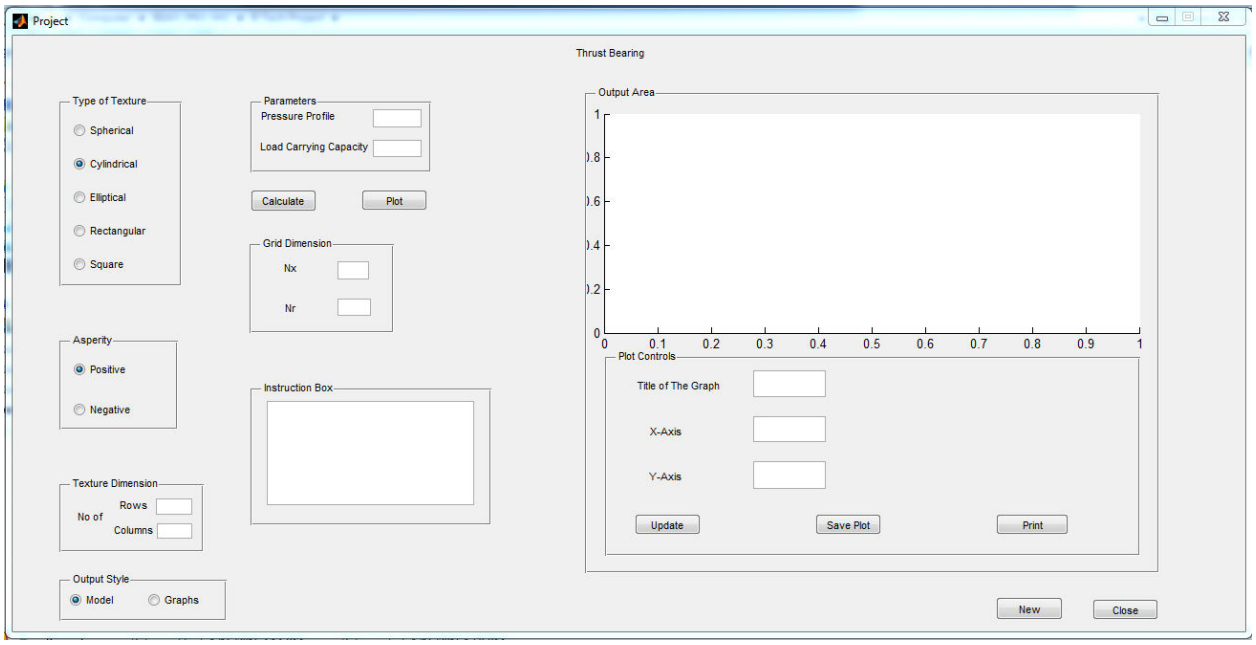


Figure-3: Iteration 3

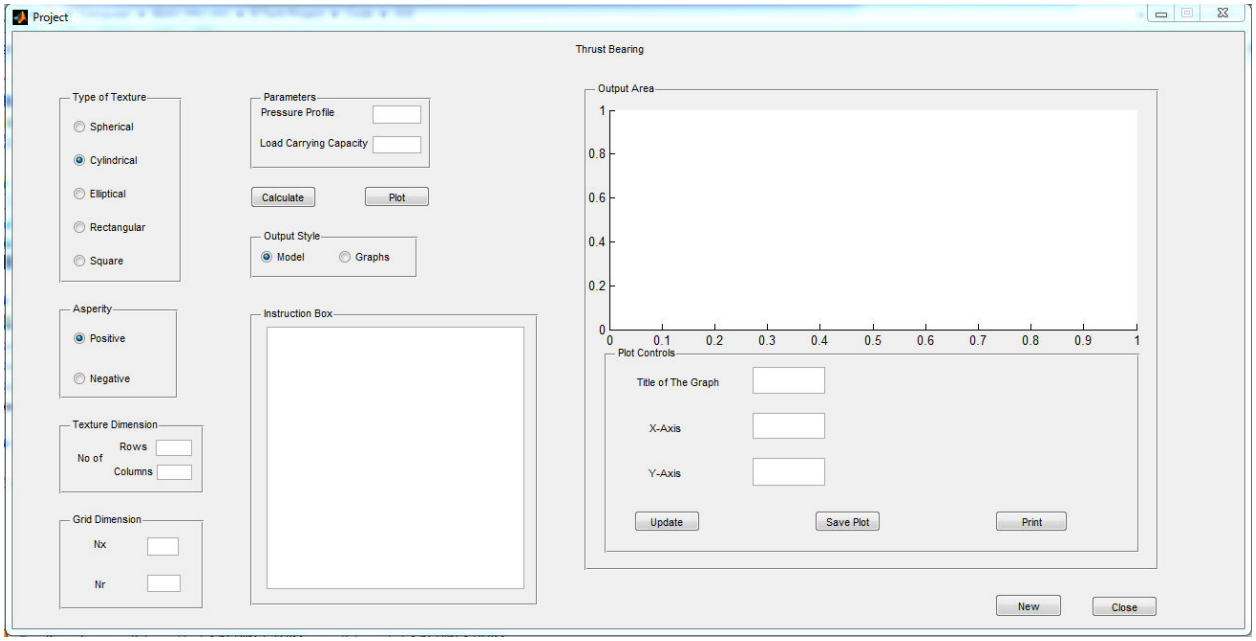


Figure-4: Iteration 4

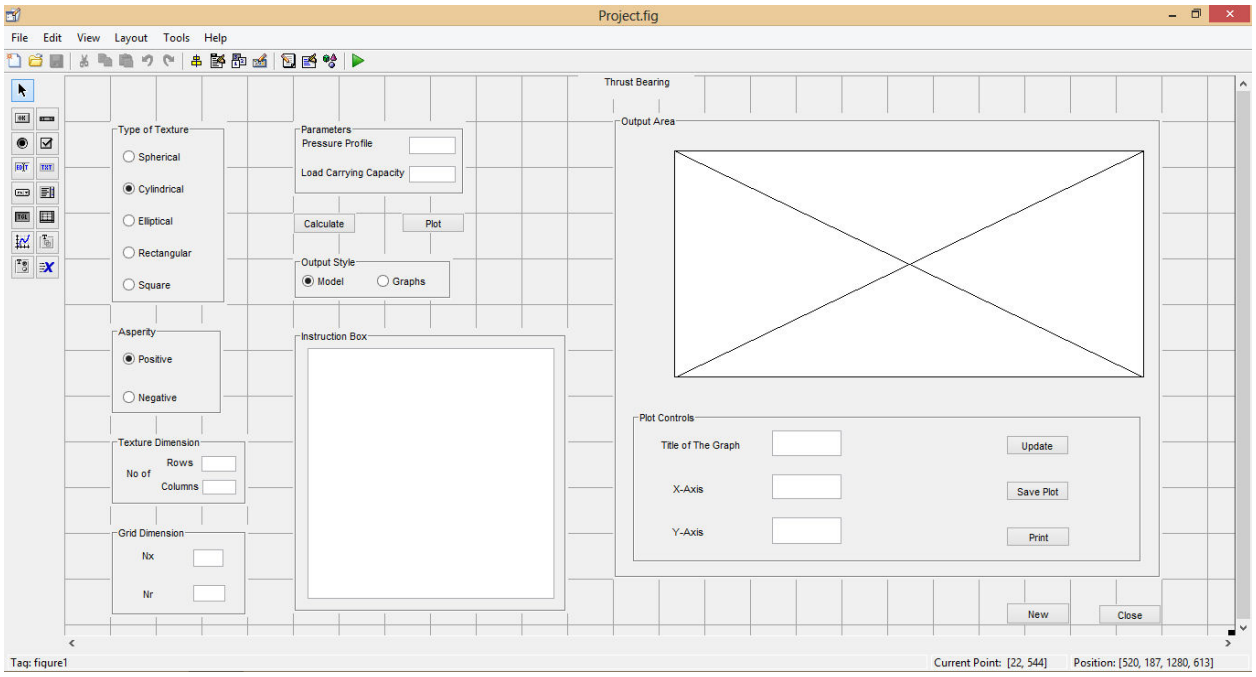
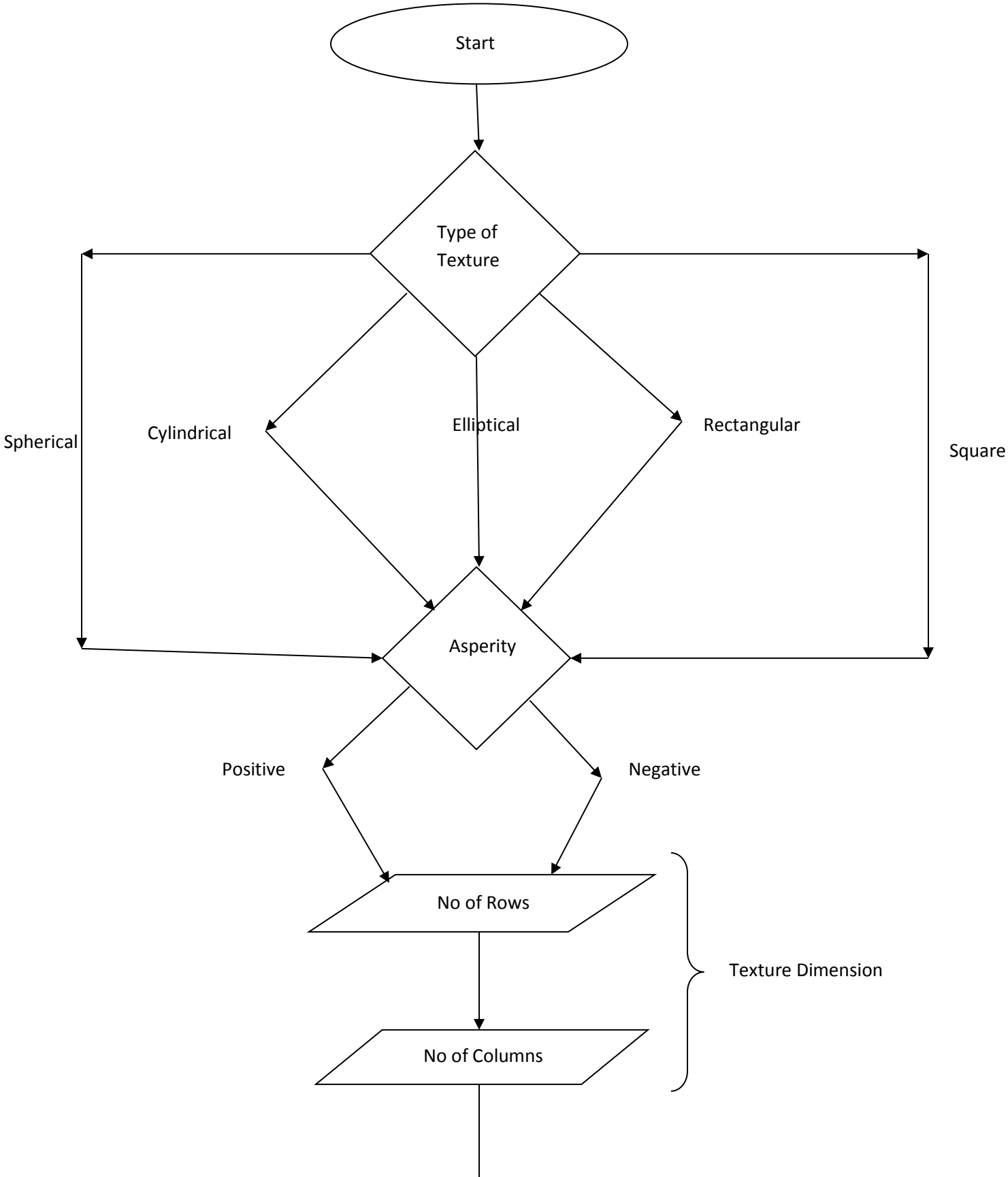


Figure-5

Figure-5: Typical Environment for Development of Software

4.4 Flow Chart



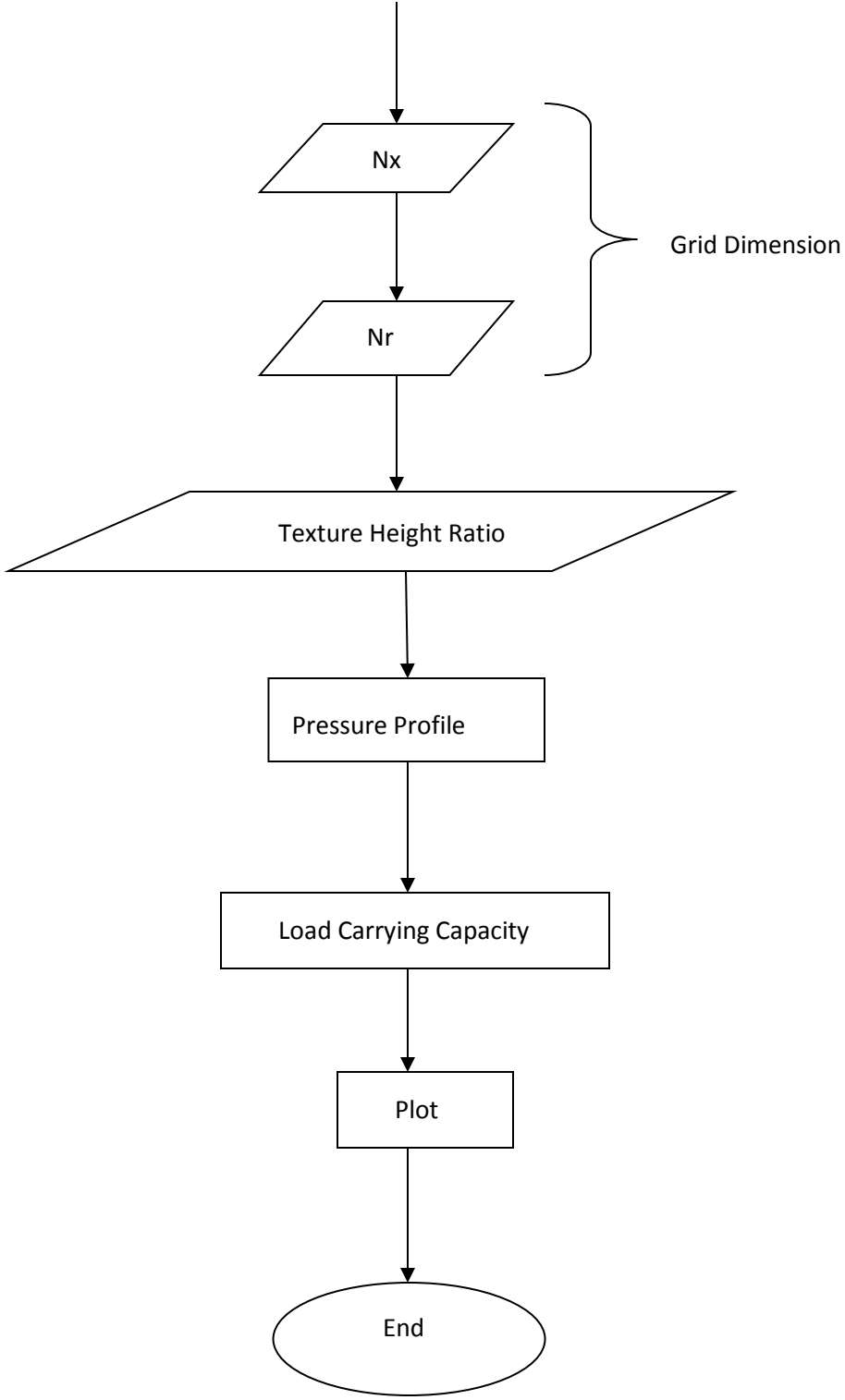


Figure-6: Flow Chart

4.5 Sample codes for different Segments

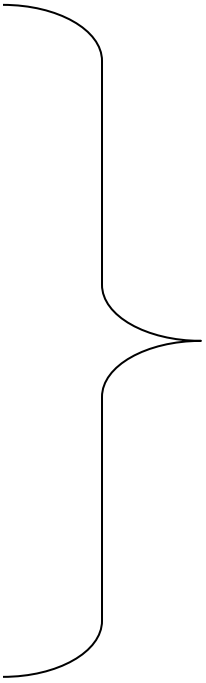
4.5.1 Sample Input

```
clc

close all

clear all

%INPUTS

ri =25;

ro =50;

N =1000;

V =0.0015;

U =2.62;

C =0.5;

omeg =(2*pi*N)/60;

Hg =0.5;

Nx =20;

Nr =20;

ang =linspace(0,2*pi,101);

ang(end) =[];

bb =0.4:0.1:1.2;

AR =0.1:0.1:8;

wb =zeros(3,8);
```

```
NR =[5,5,5,5,5];

NC =[30,40,50,60,70];

for(k=1:5)

    n_r=NR(k);

    n_c=NC(k);

for(g=1:8)

   nx=20;

   nr=20;

x =linspace(0,1,nx);

y =linspace(0,1,nr);

[X,Y] =meshgrid(x,y);
```

4.5.2 Sample for Type of texture

```
switch get(eventdata.NewValue,'Tag')  %Get Tag of Selected Object
        case 'spherical'
                display('Radio Button 1');
        case 'cylindrical'
                display(''Radio Button 2);
        case 'Elliptical'
                display(''Radio Button 3);
        case 'Rectangular'
                display(''Radio Button 4);
        case 'Square'
                display(''Radio Button 5);
end
```



Default should be spherical

Asperity

switch get(eventdata.NewValue,'Tag') %Get Tag of Selected Object

```
    case 'Positive'
        display(''Radio Button 6);
    case 'Negative'
        display(''Radio Button 7);
```

Default should be positive

end

4.5.3 Sample for Output Style

switch get(eventdata.NewValue,'Tag')%Get Tag of Selected Object

```
    case 'Model'
        display(''Radio Button 8);
    case 'Graphs'
        display(''Radio Button 9);
```

Pressure Profile

Load

As clicking on the separate radio buttons display the radio button no, so we are sure that we have exclusive selection of a button from the end button group. The output style suggests that we have correct code to go with. Instead of displaying radio button no we can have codes for respective button under their execution block and same can be executed flawlessly as shown in the demonstration.

4.5.4 Algorithm

How to activate certain portion of a code?

We can use the help of certain conditionality like for loop, while loop or if…else conditionality. Here in the project we assign different values (integer) to different textures, asperities. So the

condition will check if it satisfies or not. If it satisfies then we can do the necessary arrangement to execute that block of code only and rest will not be executed at all.

e.g. - We assign the integer ‘a’ to different textures and integer ‘b’ to different asperities. Assigned values to ‘a’ and ‘b’ are given below.

‘a’

Spherical=1

Cylindrical=2

Elliptical=3

Rectangular=4

Square=5

‘b’

Positive=0

Negative=5

Working of algorithm

c=a+b;

If c<1

c >10

Select the texture and asperity.

If c=1

{Spherical Positive}

elseif c=2

{Cylindrical Positive}

```
elseif c=3

{Elliptical Positive}

elseif c=4

{Rectangular Positive}

elseif c=5

{Square Positive}

elseif c=6

{Spherical Negative}

elseif c=7

{Cylindrical Negative}

elseif c=8

{Elliptical Negative}

elseif c=9

{Rectangular Negative}

elseif c=10

{Square Negative}

elseif c=0

{Select the Texture and Asperity}

end
```

CHAPTER 5

SOFTWARE TESTING

Testing is very much essential tom know whether a system is functioning properly or not. Through testing only we can find out the bugs and fix them for the smooth functioning of the system. It helps us realize whether we are working under specified conditions or not.

5.1 Testing Methods

Software testing cannot be done randomly and there exists a proper protocol or set of procedures to test a system fully. Software testing is of two types. They are

5.1.1 Black-box Testing

It is the method of testing when the end user or the test specialist does not have the idea or knowledge about the internal working and structures of the software. The specialist directly interacts with the User Interface of the system and tests the system putting various boundary conditions and analyzing the output variables.

It is used for very large segments of code as it is very easy to test. As the specialist does not know the back end functioning it helps in getting the viewpoint of the Customer.

5.1.2 White-box Testing

It is the method of testing when the end user or the test specialist has the idea or knowledge about the internal working and structures of the software. The specialist has the access to the source code and the testing is done by analyzing the source code with various input parameters. This kind of testing is also known as glass testing of open box testing.

Optimization of code is done with this type of testing as the specialist has access to the source code and through optimization we can remove excess lines of codes which may cause error to the system.

CHAPTER 6
RESULTS AND DISCUSSION

Different types of textures and asperity are shown in the input section and according to that different sets of output can be produced. By changing textures dimensions like no of rows and columns and also grid dimensions we can have change in the output model for our analysis.

The code is stored as .m file which gives us the scope to change it according to the demand of the future designers and to fix bugs if any at a later stage. Further a .fig file is generated which gives us the scope to realize the true front end of the program and test it further for getting the work done.

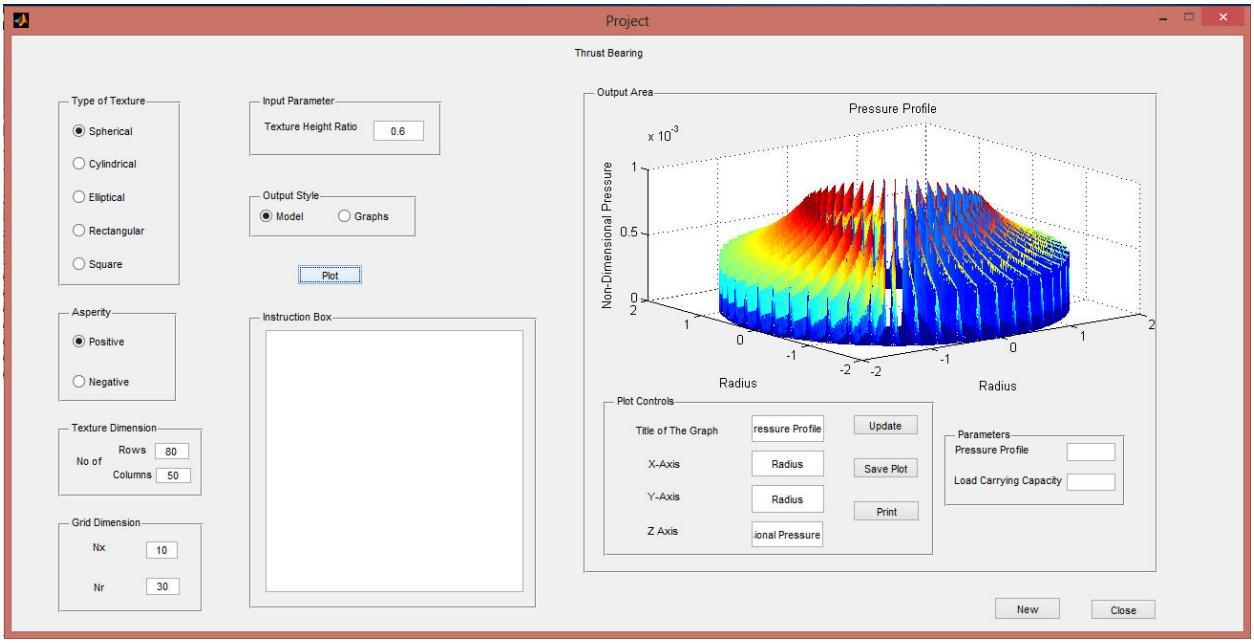


Figure-7

Test case for pressure profile of spherical positive thrust bearing (1)

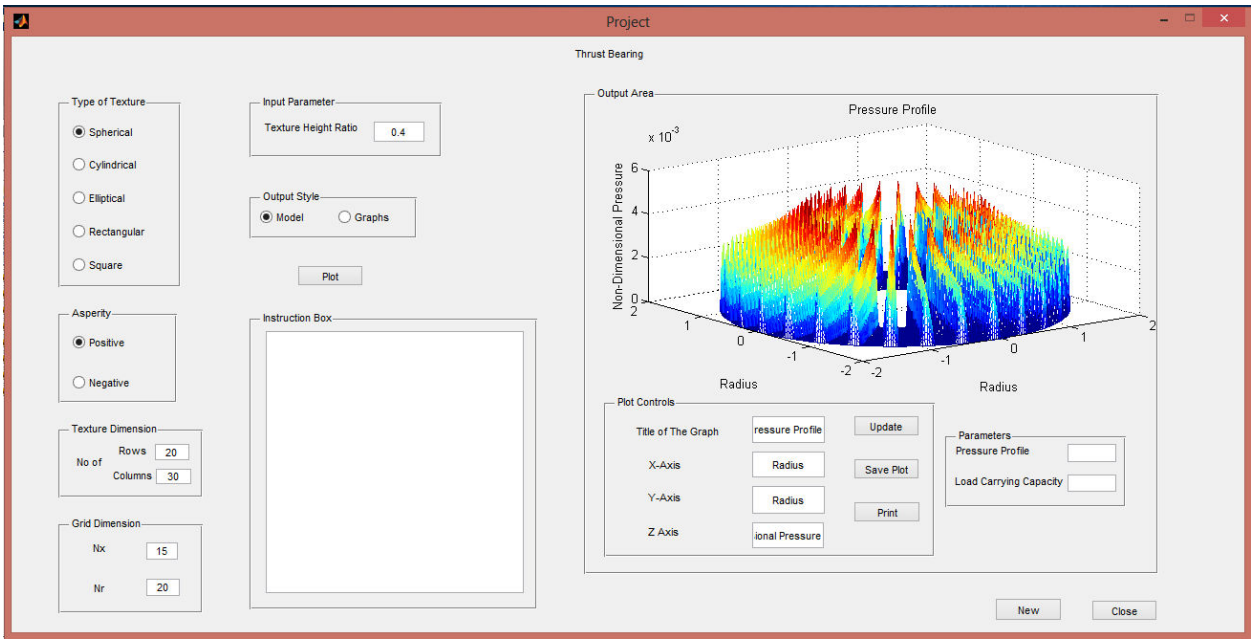


Figure-8

Test case for pressure profile of spherical positive thrust bearing (2)

Test cases of an output of a model are shown here.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Current needs to solve any engineering issue or new design need knowledge on multi-disciplinary subjects. Knowledge on all subjects is difficult to harness, so an attempt was done to reduce time for design engineers to do textured profile thrust bearing analysis. The objective of this thesis was to simplify the textured profile thrust bearing analysis of bearing to help the engineering society for faster decision making and design of modern machinery.

The software will help the design engineers and researchers to finish designing a project within a tight schedule. Further they will not have to toil hard to do the unnecessary theoretical understanding of the subject. By this they can really focus on their project to get it done error free.

Further the front end can be developed to add new features and new parameters can be added for a realistic designing of textured thrust bearing. New type of textures other than the mentioned five can be added to the features. More input variables can be placed. More output results can be shown for better understanding.

REFERENCES

[1] Sethi Durjyodhan, Hydrodynamic Analysis of Different Textured Profile Thrust Bearing, M.Tech. Thesis, NIT Rourkela, 2014

[2] Meher Siddhartha, Development of Software for Rotodynamic Analysis of Flexible Rotor, B. Tech. Thesis, NIT Rourkela, 2014

[3] Edelhofer Dr. Titus, Selecting a MATLAB Application Deployment Strategy, MathWorks

[4] Zaranek Sarah Wait, Chou Bill, Sharma Gaurav, and Zarrinkoub Houman, Accelerating MATLAB Algorithms and Applications, MathWorks

[5] Purser Rob, Deploying a Measurement Application with the MATLAB Compiler, MathWorks

[6] Cleve Moler, Simulating Blackjack with MATLAB, MathWorks

[7] Creating Graphical User Interfaces, the MathWorks, Inc (2000)

[8] www.mathworks.com